

Erledigt

Natives Powermanagement - Alternative für ssdtPRGen - Hilft ggf. bei Sleep/Wake-Problemen

Beitrag von „elmacci“ vom 24. August 2017, 19:14

Hi zusammen,

ich bin ja immer ein eifriger Leser auch anderer Foren und lese dort ab und zu über neue Möglichkeiten/Methoden für diverse Probleme die man so mit seinem Hacki haben kann (bspw. die Lösung für den iBooks Transparenzbug oder NvidiaGraphicsFixup). Wenn ich also etwas (meines Erachtens nach) spannendes habe teile ich das natürlich immer gerne mit den freundlichen Menschen hier im Forum 😊

Auch jetzt gibt es wieder so ein nette Geschichte.

Und zwar hat der in der Szene vor allem für seine HDMI/Audio-Skripte bekannte toleda eine alternative Methode für natives Powermanagement auf Skylake/Kabylake Systemen entwickelt.

Das führt dazu, dass man für das Powermanagement der CPU nicht mehr auf Pikers ssdtPRGen-Skript angewiesen ist.

Ich persönlich habe besagte Piker-SSDT bislang im Einsatz gehabt (Skylake-CPU, iMac 17,1 SMBIOS-Definition, GTX 1080TI) und hatte auch eine einwandfrei "schaltende" CPU mit unterschiedlichen Speedsteps.

Allerdings habe ich es bislang nicht geschafft, diverse Sleep/Wake Probleme vollständig zu eliminieren auf meinem System.

Insbesondere das Aufwachen meiner 3 Monitore hat in bestimmten Anwendungsszenarien nicht stabil funktioniert. Gut, ich bin da vielleicht auch ein Sonderfall - 3 4K Monitore, Wake On Lan aktiviert, Per VNC/Remote einloggen etc. 😊

Folgendes war bislang immer problematisch:

- 1) Aufwecken des Hackis über RemoteVNC
- 2) Steuerung des Hackis in der Remote-App

- 3) Remote App offen, aber keine Aktivität - am Hacki gehen die Displays aus / in den Ruhemodus
- 4) Beim wieder aufwecken (z.B. klicken auf den Desktop) crasht dann der Rechner

Jetzt könnte ich natürlich immer wieder die Remote-Verbindung trennen wenn ich gerade nichts mache - aber manchmal vergesse ich das eben und dann crasht das System.

Ihr seht - ein wirklicher Sonderfall und nicht wirklich realitätsnah 😊

Ich habe alles probiert - Grafikkarte mit SSDT-Injection, alle Darkwake Modes durch, IGPU aktiviert/deaktiviert, Primär und Sekundär, DSDT-Extraktion und Fixes, versch. [BIOS Einstellungen](#) und und und. Bis zu dem Punkt wo wirklich alle Devices die in meinem Rechner stecken bzw. nötig sind korrekt und mittels SSDT injected werden inkl. Slot-Nummer bei PCI-Devices (AppleLPC, MCHC, IMEI, LPCB, SATA Controller etc. pp)

Und jetzt zurück zum Kern des Ganzen:

Ich habe diese neue Methode einfach mal ausprobiert - und was soll ich sagen: Das oben genannte Anwendungsszenario teste ich gerade seit mehreren Stunden durch (immer wieder inaktiv per Remote, Verbindung aber steht dauerhaft. Einschlafen und Aufwecken per Remote etc.) und habe bislang noch keinen Crash gehabt 👍

Vielleicht freue ich mich auch zu früh - aber interessant ist die Methode allemal.

Und hier kurz beschrieben die Vorgehensweise (den Originalen Thread bei den Tomaten verlinke ich hier jetzt mal nicht, falls jemand mehr Hintergrundinfos braucht einfach nach "macOS Native CPU/IGPU Power Management" suchen):

Voraussetzungen:

- 1) Diese Methode funktioniert bei Skylake/KabyLake-Systemen nur, wenn auch die dazu passende, originale SMBIOS-Definition eingesetzt wird. Sprich, bei meinem Skylake 6700K also iMac 17,1.
- 2) Der IOPlatformPlugin.Kext sollte nativ/Original Apple sein. Wenn ihr da nichts explizit gemacht habt sollte das auch grundsätzlich der Fall sein.

Vorgehensweise:

1) Es muss eine XPCM SSDT in den EFI/CLover/ACPI/patched Ordner hinzugefügt und beim Start geladen werden. Diese Datei bekommt ihr direkt von toledas github hier: https://github.com/toleda/audi...r/ssdt_xcpm/ssdt_xcpm.zip

2) Falls ihr eine über ssdtPRGen generierte SSDT.aml im patched-Ordner habt, muss diese gelöscht werden da nicht mehr notwendig.

Das wars. Natives Powermanagement mit entsprechenden Speedsteps und ggf. eine Lösung für Sleep/Wake-geplagte.

Vielleicht hilft es ja dem ein oder anderen bzw. führt zum Erfolg.



cheers